strings so a configuration is not strictly necessary. The present invention does not impose the same limitations on the initial value, so a configuration is necessary when the initial value is a more complex datatype or model. The Attribute Descriptor also includes a Parent Descriptor which is a pointer to the classifier which holds the attribute descriptor as a feature.

[**0491**] The DataType Descriptor of the Attribute Descriptor includes a Base Data Name, the Constraint Descriptors, and the Initial Value Config. The DataType Descriptor of the Attribute Descriptor is also read by a DataType Handler of the Attribute Impl Generator, as indicated by connector W.

[**0492**] A Default Signal Handler of the Attribute Impl Generator creates Signal implementations of the Attribute Implementation. The default signal handler creates signals commonly generated by attributes during access and modification. The DataType Handler of the Attribute Impl Generator creates a DataType Implementation of the Attribute Implementation. The Attribute Implementation also includes a pointer to the Attribute Descriptor of which the Attribute Implementation is an implementation.

[**0493**] The Signal Implementations and DataType Implementations of the Attribute Implementation create Signal Instances and DataType Instances, respectively, of the Attribute Instance. The Attribute Instance also includes a pointer to the Attribute Implementation of which the Attribute Instance is an instance.

[**0494**] In **FIGS. 24 and 25**, a Signal is read by a Signal Handler, to create a Signal Descriptor. The Signal Descriptor is read by a Signal Impl(ementation) Generator, as indicated by connector X, to create a Signal Implementation. The signal implementation can then create Signal Instances. A Name, Documentation, Visibility, Parameters, Constraints, and Tagged Values of the Signal description are handled by a Name Handler, a Documentation Handler, a Visibility Handler, a Parameters Handler, a Constraints Handler, and a Tagged Values Handler, respectively, of the Signal Handler.

[**0495**] The Name Handler, Documentation Handler, Visibility Handler, Parameters Handler, Constraints Handler, and Tagged Values Handler of the Signal Handler map to a Name, Description, Visibility, Operations Descriptors, Constraint Descriptors, and Hint Instances, respectively, of the Signal Descriptor. The Signal Descriptor also includes a Listener type Descriptor and Role Instances.

[**0496**] The Operation Descriptors and Constraint Descriptors of the Signal Descriptor are read by an Operations Handler, as indicated by connector Y, and a Constraints Handler, as indicated by a connector Z, respectively, of the Signal Impl Generator. The Operations Handler, and the Constraints Handler of the Signal Impl Generator create Operation Implementations and Constraint Implementations, respectively, of the Signal Implementation. The Signal Implementation also includes a pointer to the Signal Descriptor of which the Signal Implementation is an implementation.

[**0497**] The Operation Implementations and Constraint Implementations of the Signal Implementation create the Operation Instances and Constraint Instances of the Signal Instance. The Signal Instance also includes a pointer to the Signal Implementation of which the Signal Instance is an instance.

[**0498**] In **FIGS. 26 and 27**, a Parameter is read by a Parameter Handler, creates a Parameter Descriptor, is read by a Parameter Impl(ementation) Generator, as indicated by connector AA, creates an implementation of a Parameter Implementation, and creates a Parameter Instance. A Name, Documentation, Parameter Kind, Constraints, type, Default Value and Tagged Values of the Parameter are handled by a Name Handler, a Documentation Handler, a Parameter Kind Handler, a Constraints Handler, a Type Handler, an Initial Value Handler, and a Tagged Values Handler, respectively, of the Parameter Handler.

[**0499**] The Name Handler, Documentation Handler, and Parameter Kind Handler map to a Name, Description, and Parameter Kind, respectively, of the Parameter Descriptor. The Constraints Handler, Type Handler, Initial Value Handler, and Tagged Value Handler of the Parameter Handler create the Constraint Descriptors of the DataType Descriptor, the Default Value Config(uration) of the DataType Descriptor, and the Hint Instances of the Parameter Descriptor The Parameter Descriptor also includes a Parent Descriptor, Multiplicity, and Role Instances and the DataType Descriptor also includes a Base Data Name.

[**0500**] The DataType Descriptor is read by a DataType Handler of the Parameter Impl Generator, as indicated by connector BB.

[**0501**] The DataType Handler of the Parameter Impl Generator creates a DataType Implementation of the Parameter Implementation. The Parameter Implementation also includes a pointer to the Parameter Descriptor of which the Parameter Implementation is an implementation.

[**0502**] The DataType Implementation of the Parameter Implementation creates a DataType Instance of the Parameter Instance. The Parameter Instance also includes a pointer to the Parameter Implementation of which the Parameter Instance is an instance.

[**0503**] In **FIGS. 28 and 29**, an Interface is read by an Interface Handler, creates an Interface Descriptor, is read by an Interface Impl(ementation) Generator, as indicated by connector CC, creates an implementation of an Interface Implementation, and creates an Interface Instance. A Name, Documentation, Visibility, Root, Leaf, Attributes, Operations, Constraints, Tagged Values, and Code Language of the Interface are handled by a Name Handler, a Documentation Handler, a Visibility Handler, a Root Handler, a Leaf Handler, an Attributes Handler, an Operations Handler, a Constraints Handler, a Tagged Values Handler, and a Language Handler, respectively, of the Interface Handler.

[**0504**] The Name Handler, Documentation Handler, Visibility Handler, Root Handler, Leaf Handler, Attributes Handler, Operations Handler, Constraints Handler, and Tagged Values Handler, map to the Name, Description, Visibility, Root, Leaf, Attribute Descriptors, Operation Descriptors, Constraint descriptors and Hint Instances, respectively, of the Interface Descriptor. The Language Handler of the Interface Handler looks up a role in the Metamodel Repository. The Language Handler uses the Metamodel Repository to lookup up a role in the Language Role that creates an instance of the language role (Role Instance<**1**>) of Role Instances of the Interface Descriptor.

[**0505**] The Attribute Descriptors, Operation Descriptors, and Constraint Descriptors are read by an Attribute Handler,